Final\_worddoc

Hannah Stroud

December 18, 2018

# Introduction

My thesis research revolves around climate justice and how to assess and evaluate the impacts of climate change adaptation on socially vulnerable populations. My field of research started when social justice advocates began noticing and proving statistically significant difference between the location of hazardous waste sites in communities of color compared to white communities (Mohai & Saha, 2006). Studies of spatial analysis supporting or refuting these claims of environmental justice differed by what unit of scale they analyzed, and what areas they included in their study (Mohai, Pellow & Roberts, 2009). Zimmerman also argues that scale plays an important role in determining injustice, as he argues that a number of scales should be explored simultaneously and a sensitivity analysis should be run to see if the scales are different in equity implications (1990). The filed has since expanded to look at various social and economic characteristics and their exposure to environmental hazards. Climate Justice looks specifically at how these vulnerable populations are affected by Climate Change and how adaptation can begin to right past inequities and injustices. Based off the work of these authors, and that fact I am still waiting to collect my own data, I conducted a social vulnerability analysis of the city of Boston at two scales and explored correlation between vulnerable populations and aspects of climate change. I conducted this analysis because I wanted to know:  
\* Where are Boston’s vulnerable populations? Does examining different levels of scale change where the most vulnerable spots are?  
\* What characteristics contribute most to those vulnerable areas?  
\* Are those socially vulnerable populations more physically exposed to climate change hazards than others?  
Social vulnerability is thought of as the product of social factors that influence the susceptibility of various groups to harm and affect their ability to harm as well as place-based inequalities like economic vitality (Cutter, Boruff & Shirley, 2003). Martin goes further and defines social vulnerability as the predisposition of social groups to suffer a disproportionately (death, injury, loss, disruption of livelihood) to hazards (2015). There are many ways, indexes and characteristics used to monitor social vulnerability. I choose seven based off their repeated mention in literature and accessibility of data from the census and Climate Ready Boston:  
\* People of Color (POC) \* Limited English Proficiency \* Low to No Income \* Medical Illness \* Disability \* Children \* Elderly  
To look at exposure and adaptability to climate hazards, I looked at areas of emergency services, open green space, daytime temperature hot spots and storm water priorities of the city. Most of my data was obtained through Climate Ready Boston and the Trust for Public Land. I used NOAA’s Social Vulnerability Index and Boston GIS for open space data.

## Setup

library(readr)  
library(tidyverse)  
library(dplyr)  
library(ggplot2)  
library(tidyr)  
library(broom)  
library(gridExtra)  
library(kableExtra)  
library(patchwork)  
theme\_set(theme\_bw())

### Data

SoVI\_neighborhood <- read.csv("./Data/CRB\_Vul\_Sum.csv")  
CRB\_Attributes <- read.csv("./Data/Climate\_Ready\_Boston\_Social\_Vulnerability.csv")  
   
#Reading in CDC composite score, uses more factors than Climate Ready Boston   
Index <- read.csv("./Data/SoVI2010\_MA.csv")  
combined <- left\_join(CRB\_Attributes, Index, by="GEOID10")  
N\_SVI <- combined %>% group\_by(Neighborhood) %>% summarise(SVI= mean(SOVI0610MA))  
write.csv(N\_SVI, file="SVI.csv")

## Methods

I used ArcMap 10.4 and R 3.5.1 to conduct my analysis and visualize my data and results. My first step was to take the social vulnerability characteristics data and find percentages since census tracts and neighborhoods were not equal. The data came to me a census tract level. There were various GIS shapefiles that had Boston neighborhoods, and all of them were different. I ended up using the neighborhood outlines from Climate Ready Boston as my boundaries for my neighborhood-scale analysis.  
Once ratios of population characteristics were computed, I calculated a Z score. I did this at the census tract and neighborhood levels. I exported that Z score as an excel and then imported it to GIS, joined it to original shape file and made maps of the Z scores for each characteristic individually for both census and neighborhood levels and displayed. I categorized the colors as consistently as I could given the range of Z score for each characteristic. Because the neighborhood level of analysis was so much larger and I had fewer data points, there was never a Z score larger than 2. I also computed a composite score of all the Z scores, to identify the most vulnerable based on all characteristics.

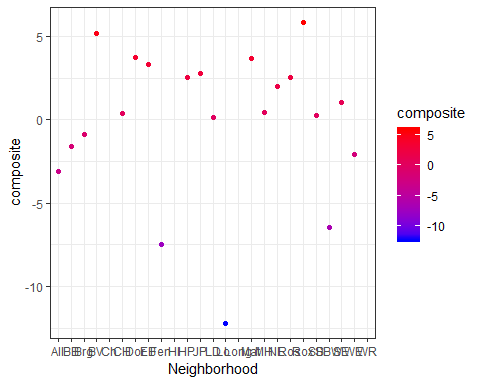
### Summary Statistics

SoVI\_neighborhood <- SoVI\_neighborhood[-c(9),] #getting rid of Harbor Island- no one lives there   
SoVI\_neighborhood <- full\_join(SoVI\_neighborhood, N\_SVI, by="Neighborhood")  
SoVI\_neighborhood <- SoVI\_neighborhood %>%   
 mutate(Per\_POC= POC2/POP100\_RE,  
 Per\_Med= MedIllnes/POP100\_RE,  
 Per\_child= TotChild/POP100\_RE,  
 Per\_eld= OlderAdult/POP100\_RE,  
 Per\_LI= Low\_to\_No/POP100\_RE,  
 Per\_Lmt\_Eng= LEP/POP100\_RE,  
 Per\_Dis= TotDis/POP100\_RE)  
#some have over 100% open space- has to do with how I got GIS data and open space that overlapped neighborhoods was counted for both   
   
CRB\_Attributes <- CRB\_Attributes[-c(151),] %>% #rid Harbod Islands again  
 mutate(Ppl\_per\_house= POP100\_RE/HU100\_RE,   
 Per\_POC= POC2/POP100\_RE,  
 Per\_Med= MedIllnes/POP100\_RE,  
 Per\_child= TotChild/POP100\_RE,  
 Per\_eld= OlderAdult/POP100\_RE,  
 Per\_LI= Low\_to\_No/POP100\_RE,  
 Per\_Lmt\_Eng= LEP/POP100\_RE,  
 Per\_Dis= TotDis/POP100\_RE)

#City averages for comparison   
Boston\_Summary <- CRB\_Attributes %>%   
 summarise(Tot\_Pop= sum(POP100\_RE),  
 Tot\_Med= sum(MedIllnes),  
 Tot\_Dis= sum(TotDis),  
 Tot\_LEP= sum(LEP),  
 Tot\_LowInc= sum(Low\_to\_No),  
 Tot\_POC= sum(POC2),  
 Tot\_Child= sum(TotChild),  
 Tot\_Eld= sum(OlderAdult)) %>%   
 mutate( Per\_POC= Tot\_POC/Tot\_Pop,  
 Per\_Med= Tot\_Med/Tot\_Pop,  
 Per\_Dis= Tot\_Dis/Tot\_Pop,  
 Per\_Child= Tot\_Child/Tot\_Pop,  
 Per\_Eld= Tot\_Eld/Tot\_Pop,  
 Per\_LI= Tot\_LowInc/Tot\_Pop,  
 Per\_LEP= Tot\_LEP/Tot\_Pop)

## Find the statistically significant difference

SoVI\_neighborhood <- SoVI\_neighborhood %>% mutate(  
 ZMed=as.numeric(scale(Per\_Med)),  
 ZChild= as.numeric(scale(Per\_child)),  
 ZEld= as.numeric(scale(Per\_eld)),  
 ZLowI= as.numeric(scale(Per\_LI)),  
 ZLEP= as.numeric(scale(Per\_Lmt\_Eng)),  
 ZDis= as.numeric(scale(Per\_Dis)),   
 ZPOC= as.numeric(scale(Per\_POC)))  
   
  
SoVI\_neighborhood <- SoVI\_neighborhood %>%   
 mutate(composite=(ZMed +ZChild+ZEld+ ZLowI +ZLEP+ZDis+ ZPOC)) %>%  
 mutate(Zcom= as.numeric(scale(composite)))  
  
write.csv(SoVI\_neighborhood, file= "SoVI\_Neighborhood\_z")  
ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=composite, color= composite)) +  
 geom\_point() + scale\_color\_gradient(low = "blue", high = "red")

 Negative scores denote neighborhoods under the average of vulnerable population characteristics. Positive denote those above average.

SoVI\_neighborhood %>%   
 select(Neighborhood, Per\_POC, Per\_Med, Per\_child, Per\_eld, Per\_LI, Per\_Lmt\_Eng, Per\_Dis) %>%  
 kable() %>%  
 kable\_styling(bootstrap\_options = "striped", full\_width = F)

Neighborhood

Per\_POC

Per\_Med

Per\_child

Per\_eld

Per\_LI

Per\_Lmt\_Eng

Per\_Dis

All

0.3966017

0.3730045

0.0227871

0.0189673

0.4346681

0.4536354

0.0526870

BB

0.2417755

0.4135978

0.0588811

0.1312775

0.1856534

0.3169309

0.0660753

BV

0.5537327

0.3956618

0.1151152

0.1576959

0.3621198

0.5198157

0.1034101

Brg

0.3250193

0.3935614

0.0697785

0.0961346

0.2806103

0.3767449

0.0904622

CH

0.2421680

0.3930574

0.2008030

0.1101649

0.2528743

0.3630391

0.0933755

Dor

0.7285745

0.3588602

0.2500036

0.0937657

0.3264079

0.4201736

0.1408853

EB

0.6283535

0.3658859

0.2138608

0.1023521

0.3380803

0.4404324

0.1278476

Fen

0.3614120

0.3714622

0.0166925

0.0182770

0.2461862

0.2644631

0.0456924

HP

0.6718458

0.3880251

0.1901869

0.1303154

0.1771028

0.3074182

0.1341121

JP

0.5265438

0.3929036

0.1727770

0.1036899

0.3073300

0.4110199

0.1081631

LD

0.4029143

0.4062335

0.0787869

0.1375805

0.2123009

0.3498814

0.0774314

Lo

0.3044641

0.3752479

0.0242748

0.0022629

0.0382637

0.0405266

0.0425838

Mat

0.7325491

0.3788479

0.2152464

0.1116126

0.2500798

0.3616924

0.1429125

MH

0.4705882

0.3750855

0.0873937

0.0675496

0.3928508

0.4604004

0.0981130

NE

0.5884270

0.3847799

0.1651388

0.1014527

0.2730714

0.3745241

0.1172534

Ros

0.5429616

0.3940977

0.2185230

0.1349942

0.1844915

0.3194857

0.1194955

Rox

0.7575920

0.3703108

0.2409095

0.1022885

0.3661427

0.4684312

0.1507582

SB

0.1929578

0.4044573

0.1565686

0.1085331

0.2488331

0.3573662

0.0971133

SBW

0.0974735

0.4099315

0.0638957

0.0743276

0.0979625

0.1722901

0.0581907

SE

0.3885102

0.3965920

0.1553067

0.0864776

0.2740993

0.3605769

0.1140458

WE

0.2042901

0.4097213

0.0681453

0.1126514

0.2060411

0.3186925

0.0688749

Ch

NA

NA

NA

NA

NA

NA

NA

HI

NA

NA

NA

NA

NA

NA

NA

Long

NA

NA

NA

NA

NA

NA

NA

WR

NA

NA

NA

NA

NA

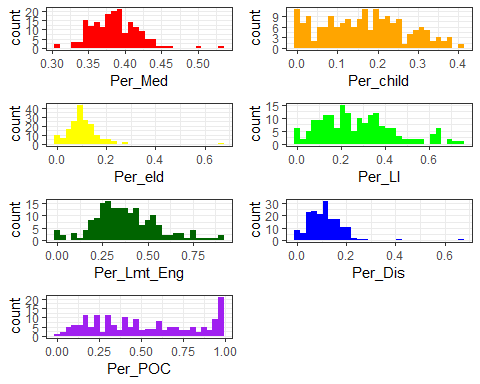
NA

NA

#Top Z scores   
Top\_NMed <- filter(SoVI\_neighborhood, ZMed >= 1.5)  
Top\_NDis <- filter(SoVI\_neighborhood, ZDis >= 1.5)  
Top\_NLEP <- filter(SoVI\_neighborhood, ZLEP >= 1.5)  
Top\_NLI <- filter(SoVI\_neighborhood, ZLowI >= 1.5)  
Top\_NEld <- filter(SoVI\_neighborhood, ZEld >= 1.5)  
  
Top\_NChild <- filter(SoVI\_neighborhood, ZChild >= 1.5)  
Top\_NPOC <- filter(SoVI\_neighborhood, ZPOC >= 1.5)  
  
Neigh\_ZTable <- rbind(Top\_NMed, Top\_NChild, Top\_NDis, Top\_NPOC, Top\_NLEP, Top\_NLI, Top\_NEld) %>%  
 select(Neighborhood\_full,ZMed, ZChild, ZDis, ZPOC, ZLEP, ZLowI, ZEld, composite) %>%  
 arrange(Neighborhood\_full) %>%  
 kable() %>%  
 kable\_styling(bootstrap\_options = "striped", full\_width = F)

Because there is a smaller range in neighborhood values, instead of going with a p-value of 0.05, I increased mine to 0.075. The p-value for Z=1.5 was found by pnorm(1.5, lower.tail=FALSE). This only applies to the characteristics that have a normal distribution: LEP, LowI, Med, Dis and eld. Their distributions are shown below.

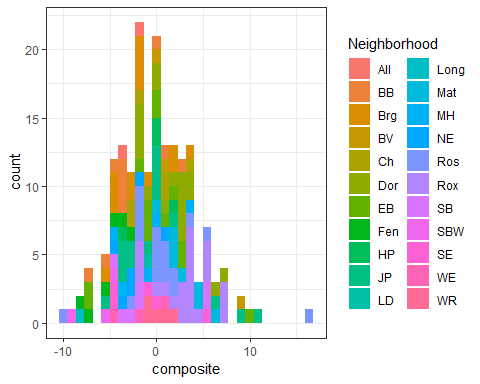
#Census Tract Leve  
CRB\_Attributes <- CRB\_Attributes %>% mutate(ZMed=as.numeric(scale(Per\_Med)),  
 ZChild= as.numeric(scale(Per\_child)),  
 ZEld= as.numeric(scale(Per\_eld)),  
 ZLowI= as.numeric(scale(Per\_LI)),  
 ZLEP= as.numeric(scale(Per\_Lmt\_Eng)),  
 ZDis= as.numeric(scale(Per\_Dis)),   
 ZPOC= as.numeric(scale(Per\_POC))) %>%  
 mutate(composite= ZMed+ ZChild + ZEld+ ZLowI+ ZLEP+ ZDis+ZPOC)  
  
#View Distribution   
med <- ggplot(CRB\_Attributes, aes(x= Per\_Med)) + geom\_histogram(fill="red")  
child <- ggplot(CRB\_Attributes, aes(x= Per\_child)) + geom\_histogram(fill="orange")  
eld <- ggplot(CRB\_Attributes, aes(x= Per\_eld)) + geom\_histogram(fill="yellow")  
lin <- ggplot(CRB\_Attributes, aes(x= Per\_LI)) + geom\_histogram(fill="green")  
lep <- ggplot(CRB\_Attributes, aes(x= Per\_Lmt\_Eng)) + geom\_histogram(fill="dark green")  
dis <- ggplot(CRB\_Attributes, aes(x= Per\_Dis)) + geom\_histogram(fill="blue")  
poc <- ggplot(CRB\_Attributes, aes(x= Per\_POC)) + geom\_histogram(fill="purple")  
  
med + child + eld + lin + lep + dis + poc + plot\_layout(ncol= 2)



combined <- left\_join(CRB\_Attributes, Index, by="GEOID10")  
ggplot(data= combined, aes(x=composite, fill= Neighborhood)) +  
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3 rows containing non-finite values (stat\_bin).



#export for GIS visualization   
write.csv(CRB\_Attributes,file= "CRB\_stats.csv")  
  
#normal distribution   
#p-value of Z 1.65 is 0.049. lower.tail=False   
Top\_Med <- filter(CRB\_Attributes, ZMed >= 1.65)  
Top\_Dis <- filter(CRB\_Attributes, ZDis >= 1.65)  
Top\_LEP <- filter(CRB\_Attributes, ZLEP >= 1.65)  
Top\_LI <- filter(CRB\_Attributes, ZLowI >= 1.65)  
Top\_Eld <- filter(CRB\_Attributes, ZEld >= 1.65)  
  
#non-normal distribution  
Top\_POC <- filter(CRB\_Attributes, ZPOC >= 1.65)  
Top\_Child <- filter(CRB\_Attributes, ZChild >= 1.65)  
  
Cen\_ZTable <- rbind(Top\_Med, Top\_Child, Top\_Dis, Top\_POC, Top\_LEP, Top\_LI, Top\_Eld) %>%   
 select(Neighborhood\_full,ZMed, ZDis, ZLEP, ZLowI, ZEld, ZPOC, ZChild) %>%   
 arrange(Neighborhood\_full) %>%   
 kable() %>%   
 kable\_styling(bootstrap\_options = "striped", full\_width = F, position = "left")

## Correlation

I obtained a Social Vulnerability Index (SVI) score from NOAA, at the census tract level. I then ran correlations between the SVI score and each of the seven characteristics to understand how much they were weighted in the SVI score. 6 of the 7 characteristics had significant correlations, though medical illness has a small negative correlation. The characteristics are important to the SVI but there are more factors they are factoring in.  
The last step of my analysis was to look at the correlation between population characteristics and exposure to climate change threats. I was able to obtain some data from the Trust from Public Land to look at urban heat island areas of concern, emergency service clusters, and storm water priorities of the city. I ran a Pearson’s correlation coefficient against the composite scores and found none of the relationships to be statistically significant. Unfortunately I could only get this information at neighborhood scale for the time frame of this project.

#Joining CRB data with CDC's   
combined <- left\_join(CRB\_Attributes, Index, by="GEOID10")  
  
ct\_p <- cor.test(combined$Per\_POC, combined$SOVI0610MA)%>% tidy() %>% mutate(name= "POC")  
  
ct\_c <- cor.test(combined$Per\_child, combined$SOVI0610MA)%>% tidy() %>% mutate(name= "Child")  
  
ct\_d <- cor.test(combined$Per\_Dis, combined$SOVI0610MA)%>% tidy() %>% mutate(name= "Dis")  
  
ct\_LI <- cor.test(combined$Per\_LI, combined$SOVI0610MA)%>% tidy() %>% mutate(name= "LowInc")  
  
ct\_LEP <- cor.test(combined$Per\_Lmt\_Eng, combined$SOVI0610MA)%>% tidy()%>% mutate(name= "LEP")  
  
ct\_m <- cor.test(combined$Per\_Med, combined$SOVI0610MA) %>% tidy() %>% mutate(name= "Med")  
 #negative correlation with percent medical illness  
ct\_eld <- cor.test(combined$Per\_eld, combined$SOVI0610MA) %>% tidy() %>% mutate(name= "Eld")  
  
rbind(ct\_p, ct\_c, ct\_d, ct\_LI, ct\_LEP, ct\_m, ct\_eld) %>%  
 kable() %>%   
 kable\_styling(bootstrap\_options = "striped", full\_width = F, position = "left") %>% column\_spec(1, bold = T, border\_right = T) %>% column\_spec(9, bold = T, border\_right = T)

estimate

statistic

p.value

parameter

conf.low

conf.high

method

alternative

name

0.4626115

6.863266

0.0000000

173

0.3374242

0.5717202

Pearson’s product-moment correlation

two.sided

POC

0.3983156

5.711674

0.0000000

173

0.2656702

0.5161602

Pearson’s product-moment correlation

two.sided

Child

0.4588607

6.792700

0.0000000

173

0.3331979

0.5685062

Pearson’s product-moment correlation

two.sided

Dis

0.5170376

7.944935

0.0000000

173

0.3993221

0.6179821

Pearson’s product-moment correlation

two.sided

LowInc

0.5322778

8.269870

0.0000000

173

0.4168490

0.6308122

Pearson’s product-moment correlation

two.sided

LEP

-0.2732241

-3.735850

0.0002538

173

-0.4051463

-0.1301564

Pearson’s product-moment correlation

two.sided

Med

0.1085112

1.435719

0.1528879

173

-0.0404836

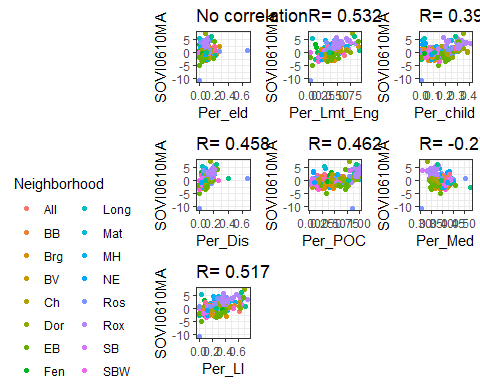
0.2527852

Pearson’s product-moment correlation

two.sided

Eld

#Viz  
Med\_plot <- ggplot(combined, aes(x= Per\_Med, y=SOVI0610MA, color= Neighborhood)) +geom\_point()+  
 ggtitle("R= -0.273")+theme(legend.position="none")  
  
Eld\_plot <- ggplot(combined, aes(x= Per\_eld, y=SOVI0610MA, color= Neighborhood)) +geom\_point()+  
 ggtitle("No correlation")+theme(legend.position="none")  
  
POC\_plot <- ggplot(combined, aes(x= Per\_POC, y=SOVI0610MA, color= Neighborhood))+   
 geom\_point() +  
 ggtitle("R= 0.462")+theme(legend.position="none")  
  
Dis\_plot <- ggplot(combined, aes(x= Per\_Dis, y=SOVI0610MA, color= Neighborhood))+   
 geom\_point() +  
 ggtitle("R= 0.458")+theme(legend.position="none")  
  
Child\_plot <- ggplot(combined, aes(x= Per\_child, y=SOVI0610MA,   
 color= Neighborhood))+   
 geom\_point() +  
 ggtitle("R= 0.398")+theme(legend.position="none")  
  
LEP\_plot <- ggplot(combined, aes(x= Per\_Lmt\_Eng, y=SOVI0610MA,   
 color= Neighborhood))+   
 geom\_point() +  
 ggtitle("R= 0.532")+theme(legend.position="none")  
  
LowInc\_plot <- ggplot(combined, aes(x= Per\_LI, y=SOVI0610MA,   
 color= Neighborhood))+   
 geom\_point() +  
 ggtitle("R= 0.517") +theme(legend.position= "left")   
  
Eld\_plot + LEP\_plot + Child\_plot+ Dis\_plot+POC\_plot+ Med\_plot+ LowInc\_plot+ plot\_layout(ncol = 3)



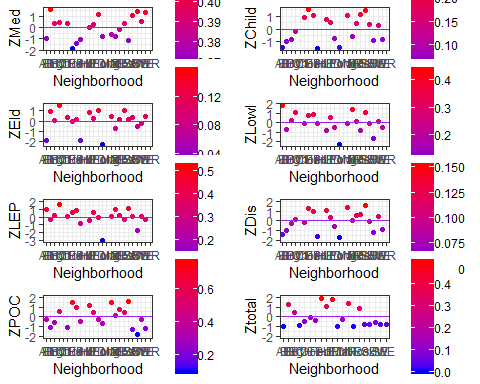
Open\_Space <- read.csv("./Data/Open\_Space\_v3.csv")  
OP\_Rec <- Open\_Space %>% filter(TypeLong== "Park\_Play\_Fields") %>%   
 group\_by(CRB\_Neighborhood) %>%   
 summarise(RecArea= as.numeric((sum(ACRES))))  
   
OP\_GS <- Open\_Space %>% filter(TypeLong== "Natural\_Areas") %>%   
 group\_by(CRB\_Neighborhood) %>%  
 summarise(Green\_area= as.numeric((sum(ACRES))))   
  
OP\_PrtAreas <-Open\_Space %>% filter(TypeLong== "Parks\_Reserv\_Beaches") %>%   
 group\_by(CRB\_Neighborhood) %>%  
 summarise(PtA\_area= as.numeric((sum(ACRES))))  
  
OP\_Garden <- Open\_Space %>% filter(TypeLong== "Community\_Gardens") %>%   
 group\_by(CRB\_Neighborhood) %>%   
 summarise(Gard\_area= as.numeric((sum(ACRES))))  
  
AllOpenSpace <- full\_join(OP\_Rec,OP\_GS, by= "CRB\_Neighborhood")  
All<- AllOpenSpace %>% full\_join(OP\_PrtAreas, by= "CRB\_Neighborhood") %>% full\_join(OP\_Garden, by="CRB\_Neighborhood")  
  
  
SoVI\_neighborhood <- SoVI\_neighborhood %>% rename(CRB\_Neighborhood= Neighborhood\_full)  
J <- full\_join(SoVI\_neighborhood, All, by= "CRB\_Neighborhood")  
J <-replace(J, is.na(J), 0)  
  
J <- J[-c(22:25),] %>%  
 mutate(Tot\_GreenSpace= RecArea + Green\_area + PtA\_area + Gard\_area) %>%  
 mutate(ratio\_green= Tot\_GreenSpace/AREA\_ACRES)  
J %>% filter(ratio\_green>= 1)

## FID AREA\_SQFT AREA\_ACRES POP100\_RE HU100\_RE TotDis TotChild OlderAdult  
## 1 13 5480700 125.8196 4861 416 207 118 11  
## Low\_to\_No LEP POC2 MedIllnes Neighborhood CRB\_Neighborhood  
## 1 186 197 1480 1824.08 Lo Longwood Medical Area  
## Shape\_\_Area Open.Space.Count Open.Space.Area.in.acres  
## 1 5.56e-05 13 182.7861  
## Storm.Water.Priorities TempHotSpotsLevel Temperature.Hot.Spots  
## 1 Low 3 High  
## Emergency.Services.Hotspots SVI Per\_POC Per\_Med Per\_child  
## 1 6 0 0.3044641 0.3752479 0.02427484  
## Per\_eld Per\_LI Per\_Lmt\_Eng Per\_Dis ZMed ZChild  
## 1 0.002262909 0.03826373 0.04052664 0.04258383 -0.8137617 -1.405866  
## ZEld ZLowI ZLEP ZDis ZPOC composite Zcom  
## 1 -2.303818 -2.318399 -2.976988 -1.663608 -0.7203732 -12.20281 -2.770902  
## RecArea Green\_area PtA\_area Gard\_area Tot\_GreenSpace ratio\_green  
## 1 71.47 0 54.9 0 126.37 1.004375

#Longwood Open Space file from GIS and neighborhood outline shows Longwood as all open space  
J <- J[-c(12),] %>% #removing longwood   
 mutate(rec\_r= RecArea/AREA\_ACRES,  
 gr\_r= Green\_area/AREA\_ACRES,   
 Pta\_r= PtA\_area/AREA\_ACRES,  
 gard\_r= Gard\_area/AREA\_ACRES) %>%   
 mutate(Zrec=as.numeric(scale(rec\_r)) ,  
 Zgreen=as.numeric(scale(gr\_r)) ,  
 ZPta= as.numeric(scale(Pta\_r)) ,  
 Zgard= as.numeric(scale(gard\_r)) ,  
 Ztotal=as.numeric(scale(ratio\_green)))  
write.csv(J, file="openspace\_joined\_census.csv" )

#Visualize   
Med<- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZMed, color =Per\_Med)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple")+ theme(legend.title=element\_blank())  
  
Child <- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZChild, color =Per\_child)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
Eld <- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZEld, color =Per\_eld)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
LowI<- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZLowI, color =Per\_LI)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
LEP <- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZLEP,ymin=-2, ymax=2, color =Per\_Lmt\_Eng)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
Dis <- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZDis,ymin=-2, ymax=2, color =Per\_Dis)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
POC <- ggplot(data= SoVI\_neighborhood, aes(x= Neighborhood, y=ZPOC,ymin=-2, ymax=2, color =Per\_POC)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
Green\_space <- ggplot(data= J, aes(x=Neighborhood, y=Ztotal,ymin=-2, ymax=2, color = ratio\_green)) + geom\_point()+scale\_color\_gradient(low = "blue", high = "red") + geom\_hline(yintercept = 0, color="purple") + theme(legend.title=element\_blank())  
  
Med + Child + Eld + LowI + LEP + Dis + POC + Green\_space + plot\_layout(ncol= 2)

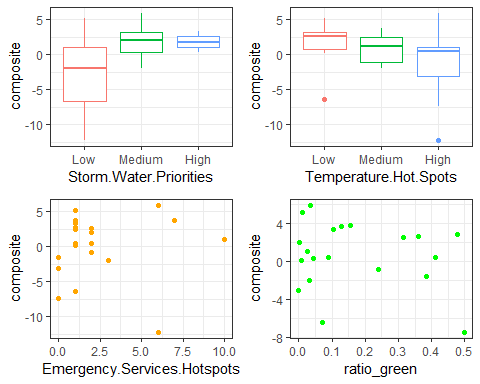
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).  
  
## Warning: Removed 4 rows containing missing values (geom\_point).



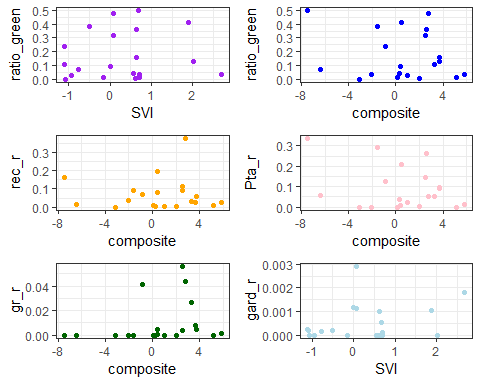
## Climate Change Impacts

Based on what data was available and easily put into R/GIS, I looked at storm water priorities of the city, temperature hot spots from heat island effect and clusters of emergency services.

SoVI\_neighborhood <- SoVI\_neighborhood[-c(22:25),] %>%   
 mutate(Storm.Water.Priorities = fct\_relevel(Storm.Water.Priorities, "Low", "Medium", "High"),  
 Temperature.Hot.Spots= fct\_relevel(Temperature.Hot.Spots, "Low", "Medium", "High"))  
  
storm <- ggplot(SoVI\_neighborhood, aes( x= Storm.Water.Priorities, y=composite, color=Storm.Water.Priorities)) +   
 geom\_boxplot() + theme(legend.position="none")  
  
em\_serv <- ggplot(SoVI\_neighborhood, aes( x= Emergency.Services.Hotspots, y=composite)) +   
 geom\_point(color= "orange")   
  
temp <- ggplot(SoVI\_neighborhood, aes( x= Temperature.Hot.Spots, y=composite,   
 color=Temperature.Hot.Spots)) +   
 geom\_boxplot() + theme(legend.position="none")  
gr\_com <- ggplot(J, aes(x= ratio\_green, y= composite)) +   
 geom\_point(color= "green")  
  
storm + temp + em\_serv + gr\_com + plot\_layout(ncol=2)



#Compare against combined characteristics and NOAA SVI correlations (total green spaces)  
gr\_sv <- ggplot(J, aes(x= SVI, y= ratio\_green)) + geom\_point(color= "purple")  
gr\_com <- ggplot(J, aes(x= composite, y= ratio\_green)) + geom\_point(color= "blue")   
  
  
#Vizualizing data   
rec\_c <- ggplot(J, aes(x= composite, y=rec\_r)) + geom\_point( color= "orange")  
rec\_sv <- ggplot(J, aes(x= SVI, y=rec\_r)) + geom\_point( color= "red")  
prt\_c <- ggplot(J, aes(x= composite, y=Pta\_r)) + geom\_point(color= "pink")  
prt\_sv <- ggplot(J, aes(x= SVI, y=Pta\_r)) + geom\_point(color= "yellow")  
green\_c <- ggplot(J, aes(x= composite, y=gr\_r)) + geom\_point(color= "dark green") #greenways and parkways  
green\_sv <- ggplot(J, aes(x= SVI, y=gr\_r)) + geom\_point(color= "green")  
gard\_c <- ggplot(J, aes(x= composite, y= gard\_r)) + geom\_point(color= "dark blue")  
gard\_c <- ggplot(J, aes(x= SVI, y= gard\_r)) + geom\_point(color= "light blue")  
  
gr\_sv + gr\_com + rec\_c +prt\_c + green\_c + gard\_c + plot\_layout(ncol=2)



cor.test(J$SVI, J$Emergency.Services.Hotspots)

##   
## Pearson's product-moment correlation  
##   
## data: J$SVI and J$Emergency.Services.Hotspots  
## t = 0.5507, df = 18, p-value = 0.5886  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.3327538 0.5404567  
## sample estimates:  
## cor   
## 0.1287213

cor.test(J$composite, J$Emergency.Services.Hotspots)

##   
## Pearson's product-moment correlation  
##   
## data: J$composite and J$Emergency.Services.Hotspots  
## t = 1.5481, df = 18, p-value = 0.139  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1175777 0.6818686  
## sample estimates:  
## cor   
## 0.342778

cor.test(J$SVI, J$TempHotSpotsLevel)

##   
## Pearson's product-moment correlation  
##   
## data: J$SVI and J$TempHotSpotsLevel  
## t = 1.2223, df = 18, p-value = 0.2374  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.188821 0.640846  
## sample estimates:  
## cor   
## 0.2768311

cor.test(J$composite, J$TempHotSpotsLevel)

##   
## Pearson's product-moment correlation  
##   
## data: J$composite and J$TempHotSpotsLevel  
## t = -0.66191, df = 18, p-value = 0.5164  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.5585675 0.3094830  
## sample estimates:  
## cor   
## -0.1541488

cor.test(J$gard\_r, J$composite)

##   
## Pearson's product-moment correlation  
##   
## data: J$gard\_r and J$composite  
## t = 1.3609, df = 18, p-value = 0.1903  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1585019 0.6589030  
## sample estimates:  
## cor   
## 0.3054427

cor.test(J$rec\_r, J$composite)

##   
## Pearson's product-moment correlation  
##   
## data: J$rec\_r and J$composite  
## t = -0.095837, df = 18, p-value = 0.9247  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.4605019 0.4241767  
## sample estimates:  
## cor   
## -0.02258313

cor.test(J$Pta\_r, J$composite)

##   
## Pearson's product-moment correlation  
##   
## data: J$Pta\_r and J$composite  
## t = -1.4796, df = 18, p-value = 0.1563  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.6736558 0.1325315  
## sample estimates:  
## cor   
## -0.3293022

cor.test(J$gr\_r, J$composite)

##   
## Pearson's product-moment correlation  
##   
## data: J$gr\_r and J$composite  
## t = 0.99276, df = 18, p-value = 0.334  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.2387505 0.6089643  
## sample estimates:  
## cor   
## 0.2278423

cor.test(J$ratio\_green, J$SVI)

##   
## Pearson's product-moment correlation  
##   
## data: J$ratio\_green and J$SVI  
## t = 0.36566, df = 18, p-value = 0.7189  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.3707392 0.5090468  
## sample estimates:  
## cor   
## 0.08586928

## Results

### Census tract vs Neighborhood

Neigh\_ZTable

Neighborhood\_full

ZMed

ZChild

ZDis

ZPOC

ZLEP

ZLowI

ZEld

composite

Allston

-0.9551668

-1.4251695

-1.3580708

-0.2502760

0.9322363

1.8305733

-1.8904032

-3.116277

Back Bay

1.6034820

-0.9568333

-0.9531888

-1.0402180

-0.3613905

-0.7757423

0.8891377

-1.594753

Bay Village

0.4729493

-0.2271674

0.1758758

0.5514258

1.5584963

1.0712459

1.5429599

5.145786

Bay Village

0.4729493

-0.2271674

0.1758758

0.5514258

1.5584963

1.0712459

1.5429599

5.145786

Dorchester

-1.8466997

1.5230759

1.3091834

1.4434899

0.6155892

0.6974668

-0.0392343

3.702871

Roxbury

-1.1249511

1.4050750

1.6077561

1.5915407

1.0722479

1.1133511

0.1716953

5.836715

Roxbury

-1.1249511

1.4050750

1.6077561

1.5915407

1.0722479

1.1133511

0.1716953

5.836715

Cen\_ZTable

Neighborhood\_full

ZMed

ZDis

ZLEP

ZLowI

ZEld

ZPOC

ZChild

Allston

-0.5390211

-1.0536775

1.3977317

2.0115657

-1.3688567

-0.4213621

-1.4715391

Back Bay

1.5656659

-0.1512463

-0.1268473

-0.9179081

1.9743756

-0.8957709

-1.0719798

Bay Village

-0.2792714

0.9451908

2.7989442

1.9602103

2.4617855

1.3912816

0.0004309

Bay Village

-0.2792714

0.9451908

2.7989442

1.9602103

2.4617855

1.3912816

0.0004309

Bay Village

-0.2792714

0.9451908

2.7989442

1.9602103

2.4617855

1.3912816

0.0004309

Charlestown

-0.6273254

0.0025745

1.0205301

1.0775647

-0.0153254

-0.1705188

2.0651074

Dorchester

-1.4841420

1.1496403

2.0398918

2.0629826

0.1981037

1.1966736

2.0625234

Dorchester

-1.7293753

0.6035994

2.2256356

2.4047586

-0.1711444

0.9371567

1.7984306

Dorchester

-0.9936749

0.7573163

0.4166338

0.4644643

-0.0680144

1.2944104

1.7965555

Dorchester

-1.4841420

1.1496403

2.0398918

2.0629826

0.1981037

1.1966736

2.0625234

Dorchester

-1.7293753

0.6035994

2.2256356

2.4047586

-0.1711444

0.9371567

1.7984306

Dorchester

-1.4841420

1.1496403

2.0398918

2.0629826

0.1981037

1.1966736

2.0625234

Dorchester

-1.7293753

0.6035994

2.2256356

2.4047586

-0.1711444

0.9371567

1.7984306

Dorchester

-1.1234947

0.0674364

1.5736860

2.0187523

-0.9221387

0.3611773

0.3526178

East Boston

2.2822553

-1.5311001

-2.1265457

-1.6238693

-1.5318160

-0.9696967

-1.5778961

East Boston

-0.4988253

1.8352601

2.8575598

2.3500298

1.6358395

0.6474786

1.4126035

East Boston

-0.4988253

1.8352601

2.8575598

2.3500298

1.6358395

0.6474786

1.4126035

East Boston

-0.4988253

1.8352601

2.8575598

2.3500298

1.6358395

0.6474786

1.4126035

Jamaica Plain

1.7531361

3.8845411

2.9016953

2.5512187

1.2462364

0.1641597

-1.5778961

Jamaica Plain

4.4275746

-1.5311001

-1.5846852

-1.6238693

-0.1004414

-1.2609628

-1.5778961

Jamaica Plain

-1.4113737

0.7650469

1.9798939

2.1660430

-0.2196861

1.2043527

1.8739959

Jamaica Plain

1.7531361

3.8845411

2.9016953

2.5512187

1.2462364

0.1641597

-1.5778961

Jamaica Plain

-0.2976560

1.7093500

2.4263916

2.1848655

0.9124215

1.3946780

1.1479182

Jamaica Plain

1.7531361

3.8845411

2.9016953

2.5512187

1.2462364

0.1641597

-1.5778961

Jamaica Plain

-1.4113737

0.7650469

1.9798939

2.1660430

-0.2196861

1.2043527

1.8739959

Jamaica Plain

-0.2976560

1.7093500

2.4263916

2.1848655

0.9124215

1.3946780

1.1479182

Jamaica Plain

1.7531361

3.8845411

2.9016953

2.5512187

1.2462364

0.1641597

-1.5778961

Jamaica Plain

-1.4113737

0.7650469

1.9798939

2.1660430

-0.2196861

1.2043527

1.8739959

Jamaica Plain

-0.2976560

1.7093500

2.4263916

2.1848655

0.9124215

1.3946780

1.1479182

Jamaica Plain

1.5161158

-0.2424283

-0.4176011

-1.1323586

1.7458795

-0.9549628

-0.1517497

North End

1.9573085

0.3419530

-0.0380694

-0.6292412

1.4826054

-1.1045466

-0.9686141

Roslindale

2.0571133

-0.0124341

-0.2521172

-1.3115823

2.6339468

-1.2712997

-0.1491421

Roslindale

3.6756508

7.4418741

1.5761677

-1.6238693

8.2492435

-1.1534184

-1.5778961

Roslindale

-0.8934184

1.0177721

0.9236643

0.8485373

0.3050269

1.1187966

1.7365251

Roslindale

3.6756508

7.4418741

1.5761677

-1.6238693

8.2492435

-1.1534184

-1.5778961

Roslindale

2.0571133

-0.0124341

-0.2521172

-1.3115823

2.6339468

-1.2712997

-0.1491421

Roslindale

3.6756508

7.4418741

1.5761677

-1.6238693

8.2492435

-1.1534184

-1.5778961

Roxbury

-1.2829152

0.9849153

1.1544884

1.3245991

-0.2830001

1.5681967

2.3218396

Roxbury

-1.6780018

0.3494874

2.5445441

2.7163259

-0.1126206

1.5431685

1.9169882

Roxbury

-1.3215353

1.3512070

1.7465930

1.9905705

-0.3944835

1.3535282

0.6087696

Roxbury

-1.6780018

0.3494874

2.5445441

2.7163259

-0.1126206

1.5431685

1.9169882

Roxbury

-1.1690605

1.4259636

1.9563866

2.0861275

-0.0807155

1.5616043

1.6425391

Roxbury

-1.3215353

1.3512070

1.7465930

1.9905705

-0.3944835

1.3535282

0.6087696

Roxbury

-0.5371596

1.3942219

1.6444834

1.6898705

0.0923452

1.3583486

1.2665029

Roxbury

-1.6780018

0.3494874

2.5445441

2.7163259

-0.1126206

1.5431685

1.9169882

Roxbury

-1.1690605

1.4259636

1.9563866

2.0861275

-0.0807155

1.5616043

1.6425391

South Boston

1.6501058

-0.8806327

-0.6655444

-0.9008996

0.5085641

-1.5171154

-0.8612916

South Boston

-1.2685869

1.1538408

1.8355250

2.1237616

-0.4946700

0.2317886

1.6922577

South Boston

-2.4104622

0.7583592

1.9957303

2.1065745

-0.0282300

0.9839840

2.0135691

South Boston

-1.2685869

1.1538408

1.8355250

2.1237616

-0.4946700

0.2317886

1.6922577

South Boston

-2.4104622

0.7583592

1.9957303

2.1065745

-0.0282300

0.9839840

2.0135691

South Boston

-1.2685869

1.1538408

1.8355250

2.1237616

-0.4946700

0.2317886

1.6922577

South Boston

-2.4104622

0.7583592

1.9957303

2.1065745

-0.0282300

0.9839840

2.0135691

South End

-0.1422573

2.1142602

1.2651445

1.4435808

-0.2900494

0.3159444

0.5996174

West End

0.8429701

-0.1073113

0.3580976

-0.3094439

1.7245081

-0.6910718

-1.2749621

### MAPS

Census Scale- Medical Illness Neighborhood Scale- Medical Illness